Simple Graph

A simple graph (or just: graph) is a tuple $\mathcal{G} = (V, E)$ where

$$V = \{A_1, \ldots, A_n\}$$

represents a finite set of vertices (or nodes) and

$$E \subseteq (V \times V) \setminus \{(A, A) \mid A \in V\}$$

denotes the set of edges.

It is called simple since there are no self-loops and no multiple edges.
Let $G = (V, E)$ be a graph. An edge $e = (A, B)$ is called

**directed** if $(A, B) \in E \Rightarrow (B, A) \notin E$
Notation: $A \rightarrow B$

**undirected** if $(A, B) \in E \Rightarrow (B, A) \in E$
Notation: $A - B$ or $B - A$

**Adjacency Set**
Let $G = (V, E)$ be a graph. The set of nodes that is accessible via a given node $A \in V$ is called the adjacency set of $A$:

$$\text{adj}(A) = \{B \in V \mid (A, B) \in E\}$$
Let $\mathcal{G} = (V, E)$ be a graph. A series $\rho$ of $r$ pairwise different nodes

$$\rho = \langle A_{i_1}, \ldots, A_{i_r} \rangle$$

is called a path from $A_i$ to $A_j$ if

$A_{i_1} = A_i$, $A_{i_r} = A_j$

$A_{i_{k+1}} \in \text{adj}(A_{i_k})$, $1 \leq k < r$

A path with only undirected edges is called an undirected path

$$\rho = A_{i_1} - \cdots - A_{i_r}$$

whereas a path with only directed edges is referred to as a directed path

$$\rho = A_{i_1} \rightarrow \cdots \rightarrow A_{i_r}$$

If there is a directed path $\rho$ from node $A$ to node $B$ in a directed graph $\mathcal{G}$ we write

$$A \xrightarrow{\rho} \mathcal{G} B.$$

If the path $\rho$ is undirected we denote this with

$$A \xleftrightarrow{\rho} \mathcal{G} B.$$
Graph Types

**Loop**

Let $\mathcal{G} = (V, E)$ be an undirected graph. A path

$$\rho = X_1 - \cdots - X_k$$

with $X_k - X_1 \in E$ is called a loop.

**Cycle**

Let $\mathcal{G} = (V, E)$ be a directed graph. A path

$$\rho = X_1 \rightarrow \cdots \rightarrow X_k$$

with $X_k \rightarrow X_1 \in E$ is called a cycle.

**Directed Acyclic Graph (DAG)**

A directed graph $\mathcal{G} = (V, E)$ is called **acyclic** if for every path $X_1 \rightarrow \cdots \rightarrow X_k$ in $\mathcal{G}$ the condition $X_k \rightarrow X_1 \notin E$ is satisfied, i.e. it contains no cycle.
Let \( G = (V, E) \) be a directed graph. For every node \( A \in V \) we define the following sets:

**Parents:**
\[
\text{parents}_G(A) = \{ B \in V \mid B \to A \in E \}
\]

**Children:**
\[
\text{children}_G(A) = \{ B \in V \mid A \to B \in E \}
\]

**Family:**
\[
\text{family}_G(A) = \{ A \} \cup \text{parents}_G(A)
\]

If the respective graph is clear from the context, the index \( G \) is omitted.

parents\( (F) = \{C, D\} \)

children\( (F) = \{J, K\} \)

family\( (F) = \{C, D, F\} \)
Let $\mathcal{G} = (V, E)$ be a DAG. For every node $A \in V$ we define the following sets:

**Ancestors:**
$$\text{ancs}_\mathcal{G}(A) = \{ B \in V \mid \exists \rho : B \xrightarrow{\rho} \mathcal{G} A \}$$

**Descendants:**
$$\text{descs}_\mathcal{G}(A) = \{ B \in V \mid \exists \rho : A \xrightarrow{\rho} \mathcal{G} B \}$$

**Non-Descendants:**
$$\text{non-descs}_\mathcal{G}(A) = V \setminus \{ A \} \setminus \text{descs}_\mathcal{G}(A)$$

If the respective graph is clear from the context, the index $\mathcal{G}$ is omitted.

\[
\begin{align*}
\text{ancs}(F) &= \{ A, B, C, D \} \\
\text{descs}(F) &= \{ J, K, L, M \} \\
\text{non-descs}(F) &= \{ A, B, C, D, E, G, H \}
\end{align*}
\]
Let \( G = (V, E) \) be a DAG.

The **Minimal Ancestral Subgraph** of \( G \) given a set \( M \subseteq V \) of nodes is the smallest subgraph that contains all ancestors of all nodes in \( M \).

The **Moral Graph** of \( G \) is the undirected graph that is obtained by

1. connecting nodes that share a common child with an arbitrarily directed edge and,
2. converting all directed edges into undirected ones by dropping the arrow heads.

Moral graph of ancestral graph induced by the set \( \{E, F, G\} \).
Let $G = (V, E)$ be an undirected graph and $X, Y, Z \subseteq V$ three disjoint subsets of nodes. We agree on the following separation criteria:

1. $Z$ u-separates $X$ from $Y$ — written as

   \[ X \perp \! \! \! \perp_G Y \mid Z, \]

   if every possible path from a node in $X$ to a node in $Y$ is blocked.

2. A path is blocked if it contains one (or more) blocking nodes.

3. A node is a blocking node if it lies in $Z$. 

\[ u\text{-Separation} \]
E.g. path $A - B - E - G - H$ is blocked by $E \in Z$. It can be easily verified, that every path from $X$ to $Y$ is blocked by $Z$. Hence we have:

$$\{A, B, C, D\} \perp_{G} \{G, H, J\} \mid \{E, F\}$$
Another way to check for u-separation: Remove the nodes in $Z$ from the graph (and all the edges adjacent to these nodes). $X$ and $Y$ are u-separated by $Z$ if the remaining graph is disconnected with $X$ and $Y$ in separate subgraphs.

$E$ separates $K$ and $B$ in the directed graph.
Lecture theatre in winter: Waiting for Mr. K and Mr. B. Not clear whether there is ice on the roads.

3 variables:

- **E** road condition: \( \text{dom}(E) = \{\text{ice, } \neg\text{ice}\} \)
- **K** K had an accident: \( \text{dom}(K) = \{\text{yes, no}\} \)
- **B** B had an accident: \( \text{dom}(B) = \{\text{yes, no}\} \)

Ignorance about these states is modelled via the observer’s belief.

\[ \xymatrix{ E \ar[d] \ar[r] & \text{influences} \quad K \quad B \\ K \ar[r] & B } \]

- (the more ice the more accidents)
- Knowledge about accident increases belief in ice
Example

<table>
<thead>
<tr>
<th>A priori knowledge</th>
<th>Evidence</th>
<th>Inferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>E unknown</td>
<td>B has accident</td>
<td>$\Rightarrow E = \text{ice}$ more likely</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Rightarrow K$ has accident more likely</td>
</tr>
<tr>
<td>$E = \lnot \text{ice}$</td>
<td>B has accident</td>
<td>$\Rightarrow$ no change in belief about $E$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Rightarrow$ no change in belief about accident of $K$</td>
</tr>
<tr>
<td>E unknown</td>
<td></td>
<td>$K$ and $B$ dependent</td>
</tr>
<tr>
<td>E known</td>
<td></td>
<td>$K$ and $B$ independent</td>
</tr>
</tbody>
</table>

Node $E$ separates $K$ and $B$ in the directed graph.
d-Separation

Now: Separation criterion for directed graphs.
We use the same principles as for u-separation. Two modifications are necessary:

- Directed paths may lead also in reverse to the arrows.
- The blocking node condition is more sophisticated.

Blocking Node (in a directed path)
A node $A$ is blocking if its edge directions along the path are of type 1 and $A \in Z$, or are of type 2 and neither $A$ nor one of its descendants is in $Z$.

- serial, head-to-tail
- serial, head-to-tail
- diverging, tail-to-tail

Type 1

- converging, head-to-head

Type 2
d-Separation

Checking path $A \rightarrow C \rightarrow E \rightarrow G$

$X$

$A \rightarrow C \rightarrow E \rightarrow G$

$Z = \{E\}$

$Y$

$B \rightarrow D \rightarrow E \leftarrow D$

$E$ is **serial** and not in $Z$: non-blocking

$E$ is also **serial** but in $Z$: **blocking**

Path is blocked, no other paths between $A$ and $G$ are available

$\Rightarrow A \perp\!\!\!\!\!\!\!\!\!\perp G \mid E$
d-Separation

Checking path $A \rightarrow C \rightarrow E \leftarrow D$:

- $C$ is \textit{serial} and not in $Z$: non-blocking
- $E$ is \textit{converging} and in $Z$: non-blocking

$\Rightarrow$ Path is not blocked

$A \not\perp D \mid E$
Checking path $A \rightarrow C \rightarrow E \leftarrow D$:

- $C$ is **serial** and not in $Z$: non-blocking
- $E$ is **converging** and not in $Z$ but one of its descendants ($J$) is in $Z$: non-blocking

$\Rightarrow$ Path is not blocked

$A \nabla D \mid J$
Checking path $A \rightarrow C \rightarrow E \leftarrow D$:

- $C$ is **serial** and not in $Z$: non-blocking
- $E$ is **converging** and not in $Z$, neither is $F, G, H$ or $J$: **blocking**

$\Rightarrow$ Path is blocked

$$A \perp\!
\perp D \mid \emptyset$$
d-Separation

Checking path $A \rightarrow C \rightarrow E \rightarrow F \rightarrow H$:

- $C$ is **serial** and not in $Z$: non-blocking
- $E$ is **serial** and not in $Z$: non-blocking
- $F$ is **serial** and not in $Z$: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp H \mid \emptyset$$
Checking path $A \to C \to E \to F \to H$:

- $C$ is **serial** and not in $Z$: non-blocking
- $E$ is **serial** and in $Z$: **blocking**
- $F$ is **serial** and not in $Z$: non-blocking

$\Rightarrow$ Path is blocked
Checking path $A \rightarrow C \rightarrow E \leftarrow D \rightarrow B$:

- $C$ is **serial** and not in $Z$: non-blocking
- $E$ is **converging** and in $Z$: non-blocking
- $D$ is **serial** and in $Z$: **blocking**

$\Rightarrow$ Path is blocked

$$A \perp\!\!\!\!\!\!\!\perp H, B \mid D, E$$
Steps

Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$. 

$Y = \{B, H\}$
Steps

Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.
Moralize that subgraph.
Steps:

Create the minimal ancestral subgraph induced by \( X \cup Y \cup Z \).

Moralize that subgraph.

Check for u-Separation in that undirected graph.

\[ A \perp \perp H, B \mid D, E \]
Example

If $C$ is not known, $A$ and $B$ are independent.

If $C$ is known, then $A$ and $B$ become (conditionally) dependent given $C$.

$A \perp B \mid C$

<table>
<thead>
<tr>
<th>Meal quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$ quality of ingredients</td>
</tr>
<tr>
<td>$B$ cook’s skill</td>
</tr>
<tr>
<td>$C$ meal quality</td>
</tr>
</tbody>
</table>
If nothing is known about the restaurant success or meal quality or both, the cook’s skills and quality of the ingredients are unrelated, that is, independent. However, if we observe that the restaurant has no success, we can infer that the meal quality might be bad.

If we further learn that the ingredients quality is high, we will conclude that the cook’s skills must be low, thus rendering both variables dependent.

\[ A \perp B \mid D \]